COMPUTING HERMITIAN DETERMINANTAL REPRESENTATIONS OF HYPERBOLIC CURVES

DANIEL PLAUMANN, RAINER SINN, DAVID E SPEYER, AND CYNTHIA VINZANT

Abstract. Every real hyperbolic form in three variables can be realized as the determinant of a linear net of Hermitian matrices containing a positive definite matrix. Such representations are an algebraic certificate for the hyperbolicity of the polynomial and their existence has been proved in several different ways. However, the resulting algorithms for computing determinantal representations are computationally intensive. In this note, we present an algorithm that reduces a large part of the problem to linear algebra and discuss its numerical implementation.

Introduction

Let f be a real homogeneous polynomial of degree d in variables x, y, z. A **Hermitian determinantal representation** of f is an expression

$$f \quad = \quad \det(xM_1 + yM_2 + zM_3), \tag{1}$$

where M_1, M_2, M_3 are Hermitian $d \times d$ matrices. The representation is **definite** if there is a point $e \in \mathbb{R}^3$ for which the matrix $e_1M_1 + e_2M_2 + e_3M_3$ is positive definite.

This imposes an immediate condition on the projective curve $\mathcal{V}_{\mathbb{C}}(f)$. Because the eigenvalues of a Hermitian matrix are real, every real line passing through e meets this hypersurface in only real points. A polynomial with this property is called **hyperbolic** (with respect to e). Hyperbolicity is reflected in the topology of the real points $\mathcal{V}_{\mathbb{R}}(f)$. When the curve $\mathcal{V}_{\mathbb{C}}(f)$ is smooth, f is hyperbolic if and only if $\mathcal{V}_{\mathbb{R}}(f)$ consists of $\lfloor \frac{d}{2} \rfloor$ nested ovals, and a pseudo-line if d is odd.

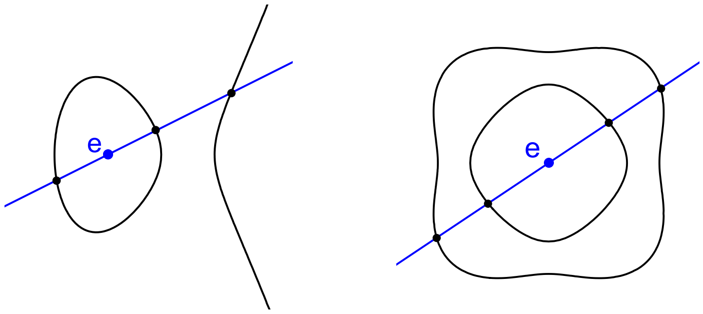

Figure 1. Cubic and quartic hyperbolic curves in $\mathbb{P}^2(\mathbb{R})$.

The Helton-Vinnikov theorem [3] (previously known as the Lax conjecture) says that every hyperbolic polynomial in three variables possesses a definite determinantal representation (1) with real symmetric matrices. Thus given a hyperbolic plane curve, one can investigate the problem of computing a definite determinantal representation.

Computing symmetric determinantal representations of hyperbolic plane curves both symbolically and numerically was investigated by Sturmfels and two of the current authors in [5] and in the case of quartic curves in [4]. Recently, it was discovered [6] that looking for Hermitian matrices M_1, M_2, M_3, rather than real symmetric matrices, greatly simplifies this computational problem and the proof of Helton and Vinnikov's theorem.

The goal of this paper is to present an algorithm for computing determinantal representations (1), examine this algorithm numerically, and to compare it with existing methods. This construction is based heavily on [6], which generalizes a classical construction due to Dixon [1].

Acknowledgements. We would like to thank Arno Fehm for helpful discussions concerning Hilbert's Irreducibility Theorem. Cynthia Vinzant was partially supported by an NSF grant DMS-1204447.

1. The Algorithm

The input to our algorithm is a polynomial $f \in \mathbb{R}[x, y, z]$ of degree d with smooth complex variety $\mathcal{V}_{\mathbb{C}}(f)$ and a point $e = (e_1, e_2, e_3) \in \mathbb{R}^3$ with respect to which f is hyperbolic. We will intersect the curve $\mathcal{V}_{\mathbb{C}}(f)$ with the degree $(d-1)$ curve given by the directional derivative

$$g(x, y, z) \quad = \quad e_1 \frac{\partial f}{\partial x} + e_2 \frac{\partial f}{\partial y} + e_3 \frac{\partial f}{\partial z}. \tag{2}$$

We assume that the intersection $\mathcal{V}_{\mathbb{C}}(f) \cap \mathcal{V}_{\mathbb{C}}(g)$ in $\mathbb{P}^2$ is transverse. That is, the two curves $\mathcal{V}_{\mathbb{C}}(f)$ and $\mathcal{V}_{\mathbb{C}}(g)$ intersect in $d(d-1)$ distinct points. In fact, this implies none of these intersection points are real [6, Lemma 2.4]. If the intersection $\mathcal{V}_{\mathbb{C}}(f) \cap \mathcal{V}_{\mathbb{C}}(g)$ is not transverse, we may replace g with the directional derivative of f in direction e', taking e' to be a generic point sufficiently close to e.

The output of the algorithm will be three Hermitian $d \times d$ matrices M_1, M_2, M_3 such that $f = c \cdot \det(xM_1 + yM_2 + zM_3)$ where $c \in \mathbb{R}_{>0}$ and $e_1 M_1 + e_2 M_2 + e_3 M_3$ is positive definite. Furthermore, g will be one of the diagonal minors of the resulting matrix $M = xM_1 + yM_2 + zM_3$, namely the minor of M obtained by removing the first row and first column from M.

The construction below is based on the idea that if the Hermitian matrix M is a determinantal representation of $f = \det(M)$, then its adjugate matrix M^{adj} satisfies

$$M^{\text{adj}} \cdot M \quad = \quad \det(M) \cdot I \quad = \quad f \cdot I.$$

Let a denote the top row of M^{adj}. Then, taking the top row of this matrix equation, we obtain the relation $a(xM_1 + yM_2 + zM_3) = (f, 0, \ldots, 0)$. Similar arguments give $(xM_1 + yM_2 + zM_3)\overline{a}^T = (f, 0, \ldots, 0)^T$. We introduce a suitable vector $a = (a_{11}, a_{12}, \ldots, a_{1d})$ and solve these linear equations in the entries of the M_i. This finds (M_1, M_2, M_3) without ever explicitly computing M^{adj}.

The algorithm proceeds as follows:

(A1) Compute the $d(d-1)$ points $\mathcal{V}_{\mathbb{C}}(f) \cap \mathcal{V}_{\mathbb{C}}(g)$.

(A2) Split the points into two disjoint, conjugate sets $\mathcal{V}_{\mathbb{C}}(f, g) = S \cup \overline{S}$.

(A3) Let a_{11} equal g.

(A4) Extend a_{11} to a basis $a = (a_{11}, \ldots, a_{1d})$ of the vector space of polynomials in $\mathbb{C}[x, y, z]_{d-1}$ that vanish on the points S. (In particular, this space is always of dimension d; see Remark 7 below.)

(A5) In the $3d^2$ variables $(M_1)_{i,j}, (M_2)_{i,j}, (M_3)_{i,j}$, solve the $2d\binom{d+2}{2} = (d+2)(d+1)d$ affine linear equations coming from the polynomial vector equations

$$\begin{aligned} a\,(xM_1 + yM_2 + zM_3) &= (f, 0 \ldots 0) \\ (xM_1 + yM_2 + zM_3)\,\overline{a}^T &= (f, 0 \ldots 0)^T. \end{aligned}$$

(A6) Output the *unique* solution M_1, M_2, M_3.

We need to argue that such a solution $M = xM_1 + yM_2 + zM_3$ exists, is unique, and has the desired properties, which we do below. Numerical implementation of this algorithm and surrounding computational issues will be discussed in Section 2.

Theorem 1. *Let $f \in \mathbb{R}[x, y, z]$ be hyperbolic with respect to a point $e \in \mathbb{R}^3$ with $f(e) > 0$. Suppose that $\mathcal{V}_{\mathbb{C}}(f)$ is smooth and that all the intersection points of $\mathcal{V}_{\mathbb{C}}(f)$ and $\mathcal{V}_{\mathbb{C}}(g)$ are transverse, where $g = e_1\frac{\partial f}{\partial x} + e_2\frac{\partial f}{\partial y} + e_3\frac{\partial f}{\partial z}$. Then the system of equations in* (A5) *has a unique solution M_1, M_2, M_3, which are Hermitian matrices satisfying*

$$f \;=\; c \cdot \det(xM_1 + yM_2 + zM_3) \qquad \text{and} \qquad e_1M_1 + e_2M_2 + e_3M_3 \succ 0,$$

where $c \in \mathbb{R}_{>0}$.

Proof. (*Existence.*) First, let us show that the affine linear equations (A5) have some solution M_1, M_2, M_3. By Construction 4.5 and Theorem 4.6 of [6], there exists a Hermitian linear matrix $M' = xM_1' + yM_2' + zM_3'$ such that for some $c \neq 0$, the determinant $\det(M')$ equals $c^{d-1}f$, the matrix $M'(e) = e_1M_1' + e_2M_2' + e_3M_3'$ is either positive or negative definite, and the first row of the adjugate matrix $A = (1/c^{d-2})(M')^{\text{adj}}$ is precisely $a = (a_{11}, a_{12}, \ldots, a_{1d})$. Since the matrices M' and $(M')^{\text{adj}}$ are Hermitian, it follows that the first column of $(1/c^{d-2})(M')^{\text{adj}}$ is $\overline{a}^T = (\overline{a_{11}}, \overline{a_{12}}, \ldots, \overline{a_{1d}})^T$.

In fact, the constant c must be positive. We can see this from examining our matrices at the point e. Since $M'(e)$ is definite, both $(M')^{\text{adj}}(e)$ and $A(e)$ must be definite as well. Furthermore, because the $(1, 1)$ entry of $A(e)$, namely $a_{11}(e) = g(e)$, is positive we see that the matrix $A(e)$ is positive definite. Then the equation

$$\det(A) \;=\; (1/c^{d-2})^d \det((M')^{\text{adj}}) = (1/c^{d-2})^d \cdot (c^{d-1}f)^{d-1} \;= c \cdot f^{d-1}$$

evaluated in the point e shows that c is positive. To find a solution to the equations (A5), let $M = (1/c)M'$. Then

$$\det(M) \;=\; (1/c)^d \det(M') \;=\; (1/c)f.$$

Furthermore M^{adj} equals $(1/c)^{d-1}(M')^{\text{adj}}$, which is $(1/c) \cdot A$. We know that both $M^{\text{adj}} \cdot M$ and $M \cdot M^{\text{adj}}$ equal $\det(M)I$. Dividing these identities by $(1/c)$ we see that $A \cdot M = f \cdot I$ and $M \cdot A = f \cdot I$. From taking the first row of the first equation and the first column of the second equation, we see that M satisfies the equations

$$a\,M = (f, 0 \ldots 0) \qquad \text{and} \qquad M\,\overline{a}^T = (f, 0 \ldots 0)^T.$$

Since $c \cdot f = \det(M)$ and $M(e)$ is positive definite, in order to finish the proof, it suffices to show that this is the unique solution to these equations.

(*Uniqueness.*) First we argue that if the system of equations (A5) has two distinct solutions, then it has two distinct solutions $M = xM_1 + yM_2 + zM_3$ and $N = xN_1 + yN_2 + zN_3$ for which both $\det(M)$ and $\det(N)$ are not identically zero.

By the proof above, there is a solution M to the equations (A5) with $\det(M) \neq 0$. If there is another solution N, then for any $\lambda \in \mathbb{R}$, the matrix $\lambda M + (1 - \lambda)N$ is also a solution. The set of $\lambda \in \mathbb{R}$ for which the determinant $\det(\lambda M + (1-\lambda)N)$ is identically zero is closed and does not contain $\lambda = 1$. Thus for almost all choices of $\lambda \in \mathbb{R}$, $\det(\lambda M + (1-\lambda)N)$ is nonzero.

Now suppose $M = xM_1 + yM_2 + zM_3$ and $N = xN_1 + yN_2 + zN_3$ are matrices satisfying the equations (A5) for which $\det(M)$ and $\det(N)$ are not identically zero. We see that at a general point point (x, y, z) in $\mathcal{V}_{\mathbb{C}}(f)$, the vector a is non-zero, and the matrix M does not have full rank. Since $\det(M)$ has degree d and f is irreducible, we can conclude that $\det(M) = \alpha f$ for some constant $\alpha \neq 0$. Similarly $\det(N) = \beta f$ for some $\beta \neq 0$.

Again we use the identity $(M)^{\mathrm{adj}} \cdot M = M \cdot (M)^{\mathrm{adj}} = \det(M)I = \alpha f I$. Therefore $(M^{\mathrm{adj}})_1 \cdot M = \alpha(f, 0 \ldots 0) = \alpha(a \cdot M)$, where $(M^{\mathrm{adj}})_1$ is the first row of M^{adj}. For generic (x, y, z) in $\mathbb{C}^3$, the matrix M is invertible, which implies that the first row of M^{adj} is αa. Similarly, the first column of M^{adj} is $\alpha \overline{a}^T$, and the first row and column of N^{adj} are βa and $\beta \overline{a}^T$, respectively.

The matrices $(1/\alpha)M^{\mathrm{adj}}$ and $(1/\beta)N^{\mathrm{adj}}$ therefore have the same first row and column and both have rank-one along the curve $\mathcal{V}_{\mathbb{C}}(f)$. For generic $(x, y, z) \in \mathcal{V}_{\mathbb{C}}(f)$, the entries of these rows and columns are non-zero. Therefore the difference $(1/\alpha)M^{\mathrm{adj}} - (1/\beta)N^{\mathrm{adj}}$ vanishes on $\mathcal{V}_{\mathbb{C}}(f)$ and all of the entries of this difference must be divisible by f. However, the entries of these matrices have degree $d-1$ whereas f has degree d, so we see that $(1/\alpha)M^{\mathrm{adj}}$ must equal $(1/\beta)N^{\mathrm{adj}}$. In particular, M^{adj} is a constant multiple of N^{adj}. It follows that M is a constant multiple of N. Because our affine linear equations (A5) are not homogenous, we see that in fact $M = N$ and our solution is unique. □

Remark 2. In fact, in our algorithm we can replace g in (2) by any polynomial $g \in \mathbb{R}[x, y, z]_{d-1}$ where $g(e) > 0$ and g **interlaces** f **with respect to** e. By this, we mean that for every point $p \in \mathbb{R}^3$ the roots of the univariate polynomial $g(te + p)$ interlace those of $f(te + p)$.

Remark 3. When the curve $\mathcal{V}(f)$ is singular, some of the above construction goes through, but requires more care. Because f has a definite determinantal representation $f = \det(M)$, e.g. [6, Cor. 4.9], we know that there exists an interlacer $g = (M^{\mathrm{adj}})_{11}$ and a basis of polynomials $a = ((M^{\mathrm{adj}})_{11}, \ldots, (M^{\mathrm{adj}})_{1d})$ for which the equations (A5) have a solution. Finding such polynomials remains a challenge. When f is singular, the intersection $\mathcal{V}(f) \cap \mathcal{V}(g)$ contains points with multiplicities. It is unclear exactly how to split these points with multiplicity and find the correct linear space of polynomials $a_{11}, \ldots, a_{1d}$ that vanish on them. One example of this procedure is given below.

Example 4. To illustrate our algorithm, we apply it to the quartic

$$f(x, y, z) \;=\; x^4 - 4x^2y^2 + y^4 - 4x^2z^2 - 2y^2z^2 + z^4, \tag{3}$$

which is hyperbolic with respect to the point $e = [1 : 0 : 0]$, and appears as Example 4.12 in [6]. This curve has two nodes, $[0 : 1 : 1]$ and $[0 : -1 : 1]$, but done carefully, our algorithm still works. Figure 2 shows the real curves $\mathcal{V}_{\mathbb{R}}(f)$ and $\mathcal{V}_{\mathbb{R}}(g)$, where $g = \frac{\partial f}{\partial x}$, in different affine planes.

First we define a_{11} to be the directional derivative $\frac{1}{4}D_e f = x^3 - 2xy^2 - 2xz^2$. The intersection of f and a_{11} consists of the eight points $[2 : \pm\sqrt{3} : \pm i]$, $[2 : \pm i : \pm\sqrt{3}]$ and the two nodes, $[0 : \pm 1 : 1]$, each with multiplicity 2. We divide these points into

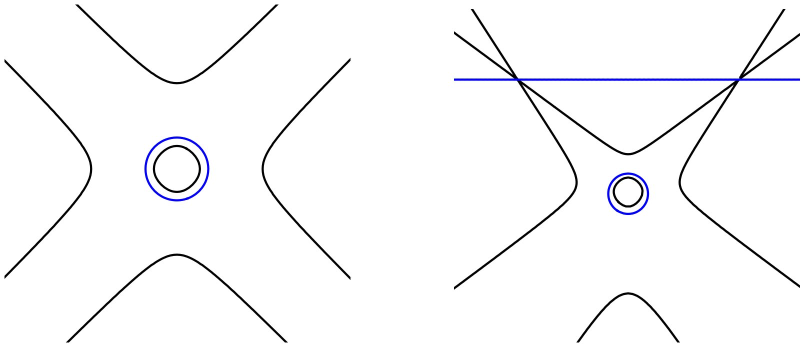

FIGURE 2. The hyperbolic quartic (3) and its directional derivative.

two conjugate sets $S \cup \overline{S}$ where

$$S = \left\{ [0:1:1],\ [0:-1:1],\ [2:\sqrt{3}:i],\ [2:-\sqrt{3}:i],\ [2:i:\sqrt{3}],\ [2:i:-\sqrt{3}] \right\}.$$

The vector space of cubics in $\mathbb{C}[x,y,z]$ vanishing on these six points is four dimensional and we extend a_{11} to a basis $\{a_{11}, a_{12}, a_{13}, a_{14}\}$ for this space, where

$$\begin{aligned} a_{12} &= ix^3 + 4ixy^2 - 4x^2z - 4y^2z + 4z^3, \\ a_{13} &= -3ix^3 + 4x^2y + 4ixy^2 - 4y^3 + 4yz^2, \\ a_{14} &= -x^3 - 2ix^2y - 2ix^2z + 4xyz. \end{aligned}$$

Let $M = xA + yB + zC$. The two 4×4 polynomial matrix equations $aM = (f, 0)$ and $M\overline{a}^T = (f, 0)^T$ give us 120 affine linear equations in the 48 variables A_{ij}, B_{ij}, C_{ij}. For example, the first entry of the vector aM is

$$\begin{aligned} &(A_{11} + iA_{21} - 3iA_{31} - A_{41})x^4 + (4A_{31} - 2iA_{41} + B_{11} + iB_{21} - 3iB_{31} - B_{41})x^3y \\ &+ (-2A_{11} + 4iA_{21} + 4iA_{31} + 4B_{31} - 2iB_{41})x^2y^2 + (-4A_{31} - 2B_{11} + 4iB_{21} + 4iB_{31})xy^3 \\ &- 4B_{31}y^4 + (-4A_{21} - 2iA_{41} + C_{11} + iC_{21} - 3iC_{31} - C_{41})x^3z \\ &+ (4A_{41} - 4B_{21} - 2iB_{41} + 4C_{31} - 2iC_{41})x^2yz + (-4A_{21} + 4B_{41} - 2C_{11} + 4iC_{21} + 4iC_{31})xy^2z \\ &+ (-4B_{21} - 4C_{31})y^3z + (-2A_{11} - 4C_{21} - 2iC_{41})x^2z^2 + (4A_{31} - 2B_{11} + 4C_{41})xyz^2 \\ &+ (4B_{31} - 4C_{21})y^2z^2 + (4A_{21} - 2C_{11})xz^3 + (4B_{21} + 4C_{31})yz^3 + 4C_{21}z^4. \end{aligned}$$

Identifying this polynomial with f gives us 15 affine linear equations. For example, from the monomial x^4, we see that $A_{11} + iA_{21} - 3iA_{31} - A_{41} = 1$. Similarly, from each of the other entries of $aM - (f, 0)$ and $M\overline{a}^T - (f, 0)^T$ we get 15 affine linear equations in the $3 \cdot 4^2$ variables A_{ij}, B_{ij}, C_{ij}, for a total of $2 \cdot 4 \cdot 15 = 120$. The unique solution to these 120 equations gives the Hermitian matrix representation

$$xA + yB + zC \quad = \quad \frac{1}{8} \begin{pmatrix} 14x & 2z & 2ix - 2y & 2i(y - z) \\ 2z & x & 0 & -ix + 2y \\ -2ix - 2y & 0 & x & ix - 2z \\ -2i(y - z) & ix + 2y & -ix - 2z & 4x \end{pmatrix},$$

whose determinant is $(1/256) \cdot f$.

Ideally we would like to carry out our algorithm symbolically, as in this example, but the required field extensions generally will be too large. Given a hyperbolic polynomial $f \in \mathbb{Q}[x, y, z]_d$, one can ask: What is the field extension necessary to carry out the above construction symbolically?

In fact, after computing the points $\mathcal{V}(f, g)$ and splitting them as $S \cup \overline{S}$, all of the remaining steps in the algorithm only require linear algebra, and thus can be done with rational arithmetic. Unfortunately, for generic $g \in \mathbb{R}[x, y, z]_{d-1}$ the Galois group of the intersection $\mathcal{V}(f, g)$ is the full symmetric group of size $d(d-1)$. It seems very hard to say anything about the smallest field extension of the points $\mathcal{V}(f, g)$ for special choices of the interlacing polynomial g.

Remark 5. Since the primary computational task is to find the coordinates of the points in $\mathcal{V}(f, g)$, it would be worth investigating whether there is a way to find the polynomials a_{1j} without computing the points S explicitly. The analogous 1-dimensional problem is, given a positive real polynomial $f(t)$ of degree $2d$, to write $f(t)$ as $g(t)\overline{g}(t)$ for some degree d polynomial $g \in \mathbb{C}[t]_{\leq d}$. This task is known as "polynomial spectral factorization", see, for example, [2]. Both problems involve finding a basis of polynomials (in either $\mathbb{C}[t]_{\leq d}$ or $\mathbb{C}[x, y, z]_{d-1}/(f)$) that vanish on a set of non-real points, given a real polynomial that vanishes on both the set and its complex conjugate. This higher dimensional analogue does not seem to have been considered, but strikes us as worthy of study.

2. Numerical Implementation

Here we discuss our numerical implementation of this algorithm in `Mathematica` which is available as supplementary material on the website `http://people.math.gatech.edu/~rsinn3/numdetrep.html`. Below we give computation times and an error measure for 100 random tests for polynomials of degrees 3-15. Overall, this method results in very fast computations compared to other available methods, although the accuracy becomes poor for large d. The numerical accuracy of the approximation of the common zeros of the hyperbolic polynomial f and the interlacing polynomial g (step (A1)) strongly affects the final error in the numerical determinantal representation. The following steps (A4) and (A5) in our algorithm are all linear algebra problems that can be solved accurately.

One issue with numerical computations is that the affine linear equations in (A5) are overdetermined – there are $d^3 + 3d^2 + 2d$ equations in $3d^2$ variables. With small numerical errors, these equations no longer have a solution. In our implementation of the algorithm, we solve this by the standard method of taking a least squares solution to the system.

The step that takes the most significant amount of computation time by far is computing the points $\mathcal{V}(f, g)$. The following steps in our algorithm

- Computing the basis $(a_{11}, \ldots, a_{1d})$.
- Translating the polynomial equations (A5) into a system of linear equations.
- Solving the resulting least squares problem.

take much less time in our implementation.

Overall this method finds (approximate) determinantal representations surprisingly fast. To test our code, we generated hyperbolic polynomials of degree d by taking the determinant of $xI + y(B + B^T) + z(C + C^T)$ where B and C are random $d \times d$ matrices whose entries are chosen randomly from a normal distribution with mean 1 and standard deviation 0.5. Any such determinant will be hyperbolic with

degree	**3**	**4**	**5**	**6**	**7**	**8**	**9**
time (sec)	0.05	0.11	0.25	0.56	1.19	2.36	4.47
time $\mathcal{V}(f,g)$	0.04	0.09	0.20	0.45	0.96	1.89	3.58
error	$5 \cdot 10^{-13}$	$3 \cdot 10^{-12}$	$2 \cdot 10^{-11}$	$2 \cdot 10^{-10}$	$4 \cdot 10^{-9}$	$4 \cdot 10^{-8}$	$1 \cdot 10^{-7}$
relative error	$2 \cdot 10^{-14}$	$9 \cdot 10^{-14}$	$2 \cdot 10^{-13}$	$4 \cdot 10^{-13}$	$2 \cdot 10^{-12}$	$1 \cdot 10^{-11}$	$9 \cdot 10^{-12}$

degree	**10**	**11**	**12**	**13**	**14**	**15**
time (sec)	9.04	13.81	22.95	37.06	58.08	89.71
time $\mathcal{V}(f,g)$	6.49	11.20	18.72	30.37	47.84	74.31
error	$2 \cdot 10^{-6}$	0.03	$6 \cdot 10^{-4}$	0.01	5.89	47827
relative error	$6 \cdot 10^{-11}$	$3 \cdot 10^{-7}$	$5 \cdot 10^{-10}$	$4 \cdot 10^{-9}$	$5 \cdot 10^{-7}$	$1 \cdot 10^{-4}$

Table 1. Average times and errors for computing determinantal representations.

respect to the point $[1:0:0]$. Averaging test times for 100 examples in each degree gave the computation times shown in Table 1.

Here "error" means the maximum over the absolute values of the coefficients of the difference between the original polynomial f and the appropriately scaled determinant $c \cdot \det(M)$. We also found it useful to look at the "relative error", meaning the error divided by the largest coefficient of f. This is a more reasonable measure for the accuracy in the coefficients of a hyperbolic polynomial because they tend to become very large in higher degree. Indeed, differences between the coefficients of a hyperbolic polynomial can grow exponentially with the degree, as in [7, §4].

One additional source of numerical errors is the computation of the determinant of the output of our algorithm. Because of the size of this matrix, a symbolic computation of the determinant is infeasible and instead we compute it by interpolation on a random set of points on the unit circle. Then we use the interpolated polynomial to compute the errors in the coefficients. Numerical testing suggests that numerical error in the coefficients of the interpolated determinant are small.

For comparison, the only other known methods for computing definite determinantal representations are discussed in [5]. Here, finding definite determinantal representations is already extremely time consuming for quintics ($d = 5$) and practically infeasible for larger degrees ($d \geq 6$). Thus the method described above provides a great improvement in computation ability.

Remark 6. As we use a random matrix of linear forms to generate our test polynomials, one might also think to compare the starting matrices to those in the output, rather then the coefficients of the determinants. The reason this cannot be done is that a given hyperbolic polynomial f has an infinite collection of Hermitian determinantal representations $f = \det(M)$, even when considered up to the equivalence $M \sim U^*MU$ for $U \in GL(d, \mathbb{C})$. There is no reason for the starting determinantal representation $xI + y(B + B^T) + z(C + C^T)$ and the output $xM_1 + yM_2 + zM_3$ to be equivalent, and so we cannot meaningfully compare their entries.

Remark 7. The algorithm presented here could also be used to construct a Hermitian determinantal representation of a non-hyperbolic polynomial f, under certain conditions. If $f \in \mathbb{R}[x, y, z]_d$, then one would need a polynomial $g \in \mathbb{R}[x, y, z]_{d-1}$

such that the intersection of the two curves $\mathcal{V}_{\mathbb{C}}(f) \cap \mathcal{V}_{\mathbb{C}}(g)$ is transversal and contains no real points. Furthermore, the vanishing ideal of S in step (A4) of our algorithm must have dimension d in degree $d-1$; this translates into a condition on the corresponding divisor on $\mathcal{V}_{\mathbb{C}}(f)$ (see [9, §4.5]). It is a non-trivial fact that this condition is always met in our original setup when f is hyperbolic and g interlaces f. This follows from [9, Thm. 5.4] or the main result of [6]; a fuller discussion of the non-degeneracy condition in terms of points on the Jacobian can be found in [8]. In the non-hyperbolic case, the condition is at least met for a generic choice of $g \in \mathbb{R}[x, y, z]_{d-1}$. If d is odd, then the reality condition can be obtained by picking g to have no real points, for example by taking g as a sum of squares. Otherwise it is not clear how to pick such a polynomial g.

References

[1] Alfred Cardew Dixon. Note on the reduction of a ternary quantic to a symmetrical determinant. *Math. Proc. Cambridge Phil. Society*, 11:350–351, 1902.

[2] Tim N. T. Goodman, Charles A. Micchelli, Giuseppe Rodriguez, and Sebastiano Seatzu. Spectral factorization of Laurent polynomials. *Adv. Comput. Math.*, 7(4):429–454, 1997.

[3] J. William Helton and Victor Vinnikov. Linear matrix inequality representation of sets. *Comm. Pure Appl. Math.*, 60(5):654–674, 2007.

[4] Daniel Plaumann, Bernd Sturmfels, and Cynthia Vinzant. Quartic curves and their bitangents. *J. Symbolic Comput.*, 46(6):712–733, 2011.

[5] Daniel Plaumann, Bernd Sturmfels, and Cynthia Vinzant. Computing linear matrix representations of Helton-Vinnikov curves. In *Mathematical methods in systems, optimization, and control*, volume 222 of *Oper. Theory Adv. Appl.*, pages 259–277. Birkhäuser/Springer Basel AG, Basel, 2012.

[6] Daniel Plaumann and Cynthia Vinzant. Determinantal representations of hyperbolic plane curves: an elementary approach. *J. Symbolic Comput.*, 57:48–60, 2013.

[7] David E. Speyer. Horn's problem, Vinnikov curves, and the hive cone. *Duke Math. J.*, 127(3):395–427, 2005.

[8] Victor Vinnikov. Selfadjoint determinantal representations of real plane curves. *Math. Ann.*, 296(3):453–479, 1993.

[9] Victor Vinnikov. LMI representations of convex semialgebraic sets and determinantal representations of algebraic hypersurfaces: past, present, and future. In *Mathematical methods in systems, optimization, and control*, volume 222 of *Oper. Theory Adv. Appl.*, pages 325–349. Birkhäuser/Springer Basel AG, Basel, 2012.

Universität Konstanz, Germany
E-mail address: Daniel.Plaumann@uni-konstanz.de

Georgia Institute of Technology, Atlanta, GA, USA
E-mail address: rsinn3@math.gatech.edu

University of Michigan, Ann Arbor, MI, USA
E-mail address: speyer@umich.edu

North Carolina State University, Raleigh, NC, USA
E-mail address: clvinzan@ncsu.edu